Communication

Pix2Pix and Deep Neural Network-Based Deep Learning Technology for Predicting Vortical Flow Fields and Aerodynamic Performance of Airfoils

Han-Seop Song, Jophous Mugabi * **and Jae-Ho Jeong ***

Thermal-Fluid Energy Machine Lab., Department of Mechanical Engineering, Gachon University, 1342, Seongnam-daero, Sujeong-gu, Seongnam-si 13306, Republic of Korea
* Correspondence: mugajoph@gachon.ac.kr (J.M.); jaeho.jeong@gachon.ac.kr (J.-H.J.); Tel./Fax: +82-31-750-5654 (J.-H.J.)

Abstract: Traditional computational fluid dynamics (CFD) methods are usually used to obtain information about the flow field over an airfoil by solving the Navier–Stokes equations for the mesh with boundary conditions. These methods are usually costly and time-consuming. In this study, the pix2pix method, which utilizes conditional generative adversarial networks (cGANs) for image-to-image translation, and a deep neural network (DNN) method were used to predict the airfoil flow field and aerodynamic performance for a wind turbine blade with various shapes, Reynolds numbers, and angles of attack. Pix2pix is a universal solution to the image-to-image translation problem that utilizes cGANs. It was successfully implemented to predict the airfoil flow field using fully implicit high-resolution scheme-based compressible CFD codes with genetic algorithms. The results showed that the vortical flow fields of the thick airfoils could be predicted well using the pix2pix method as a result of deep learning.

Keywords: pix2pix; image-to-image translation; airfoil; deep learning; GAN; DNN; wind turbine blade; CFD

1. Introduction

Globally, the demand for renewable clean energy sources is growing rapidly. Wind energy is one of the most technologically advanced and fastest-growing sustainable energy industries. In the Annual Wind Report, it was estimated that about 93.6 GW of capacity was installed in the year 2021 [1]. Although about 1.8% lower than the year 2020, the overall installed wind capacity rose to 837 GW, an increase of 12.4%. However, it is estimated that, for the world to maintain a global temperature increase below 1.5 °C and attain net zero emissions by 2050, the wind energy growth rate needs to quadruple by the end of the decade [1].

One of the strategies adopted by the wind energy industry is to increase the size of the wind turbine blades so that more energy is captured from the wind, especially in offshore installments where high wind speeds offer the potential for large energy capture [2,3]. Large wind turbines with long, slender, and flexible blades enhance the aerodynamic performance, thereby increasing the annual energy production (AEP) and decreasing the cost of energy (COE) of wind farms. However, longer blades increase the design load of the blades and the entire wind turbine system. Therefore, to withstand the increased load while maintaining the aerodynamic performance of the blades, the optimal design and placement of airfoils in the spanwise direction are two of the most important aspects of blade design [4]. Rotor blades are designed based on a combination of several airfoils with different thickness values depending on their spanwise position on the blade [5]. To minimize the aerodynamic load on the rotor blade as it becomes longer, the outboard rotor blade is usually made sharper, which increases the blade root bending moment. In addition, the airfoil thickness

of the inboard rotor blade is made thicker to maximize the sectional moment of inertia of the thin-shelled airfoil structure [4]. Multi-objective optimization should thus be conducted during the design process for an airfoil since it involves several aerodynamic requirements, such as high lift, low drag, and stall characteristics [5,6]. These aerodynamic requirements can be determined through flow field analysis using computational fluid dynamics (CFD) simulations by solving the Navier–Stokes equations for the mesh with boundary conditions. However, CFD simulations in the airfoil design process require a lot of time and expensive computation [7–9].

Recently, data-driven approaches, such as machine learning and deep learning, have received considerable attention in the field of fluid dynamics due to the powerful learning capabilities of neural networks [8,10,11]. After training, the neural network model can be used to obtain the prediction results for the airfoil flow field in a few seconds or even milliseconds. This provides a faster alternative to CFD simulations as an efficient function-approximation technique in high-dimensional spaces. Deep learning has been used for the prediction of airflow in several studies. Bhatnagar et al. [12] proposed an approximation model based on convolution neural networks (CNNs) to predict the velocity and pressure field of new geometries under new flow conditions. Data from the Reynolds-averaged Navier–Stokes (RANS) flow solutions for flow over airfoil shapes were used to train the model. The trained model effectively detected essential features from new geometries with minimal supervision and could effectively estimate the velocity and pressure field much faster, which made it possible to study the impacts of the airfoil shape and operating conditions on the aerodynamic forces and the flow field in near-real time. Sekar et al. [7] used a combination of a CNN and a multilayer perceptron (MLP) network to predict the incompressible laminar steady flow field over airfoils. The CNN was employed to extract the geometrical parameters from airfoil shapes and the results were fed as input into the MLP network to obtain an approximate model to predict the flow field. The CNN could efficiently and accurately estimate the entire velocity field two to four orders of magnitude faster than the CFD solver and with a lower error rate [9]. Recently, the data-augmented generative adversarial network (GAN) model has gained attention for its rapid and accurate flow field prediction [13]. The GAN can be adapted to the task with sparse data and can learn losses by attempting to determine whether the output image is real or fake while simultaneously training a generative model to minimize this loss. In this way, an output indistinguishable from reality can be obtained, unlike with CNNs, which tend to minimize the Euclidean distance between the predicted and ground-truth pixels, leading to the production of blurry results [14]. Conditional generative adversarial networks (cGANs), an extension of the GAN model that enables the model to be conditioned with external information, have also been studied [14,15]. cGANs are suitable for image-to-image translation tasks where an input image is conditioned to generate a corresponding output image. When a cGAN is combined with a U-Net architecture, a mapping relationship between the geometry shape and flow field can be established and good prediction results with large-scale test sets can be obtained [8].

In this study, we developed an airfoil flow field and aerodynamic performance prediction model that uses deep-learning technology instead of CFD simulation. Among the various deep-learning models, the pix2pix method [14] for image-to-image transformation and the deep neural network (DNN) method were selected. The pix2pix method, a universal solution to the image-to-image translation problem that utilizes cGANs [14], was implemented to predict the airfoil flow field. In addition, the DNN method was implemented to predict the airfoil aerodynamic performance coefficient. A dataset obtained using an in-house CFD code with a genetic algorithm was used to train the pix2pix and DNN models.

2. Methods

This section describes the deep-learning techniques used to predict the airfoil flow field and the data used to train the deep learning.

2.1. Generative Adversarial Network (GAN)

The generative adversarial network (GAN) is a generative model and one of the most active research topics in the field of deep learning [13]. The GAN architecture consists of a generator and discriminator, which generate data through adversarial training. The generator G produces fake data from random vector noise and the discriminator D distinguishes between real and fake data. The generator is trained to generate data that the discriminator cannot distinguish from real data, and the discriminator is trained to accurately distinguish fake data from real data. The architecture of the GAN is shown in Figure 1a.
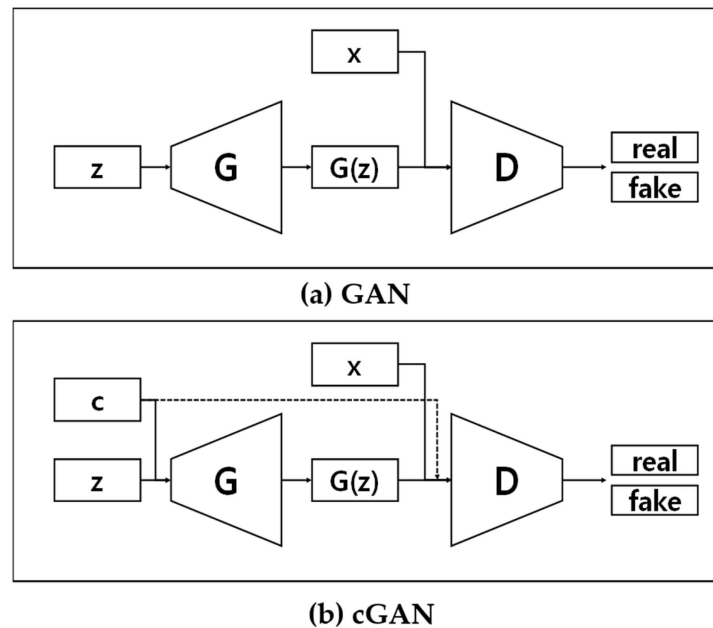

(a) GAN

(b) cGAN

Figure 1. Architectures of (**a**) the generative adversarial network (GAN) and (**b**) conditional generative adversarial network (cGAN).

2.2. Conditional Generative Adversarial Network (cGAN)

The cGAN is a variant of the GAN that was proposed to conditionally generate data [16]. The cGAN conditions can be input in various forms, such as noise vectors, images, and class labels. The architecture of the cGAN is shown in Figure 1b, where the input z and condition c are combined and provided to the generator G. The input x to the discriminator is also provided and combined with the condition c.

2.3. Image-to-Image Translation with Conditional Adversarial Net (Pix2pix)

Pix2pix is a universal solution to the image-to-image translation problem that utilizes cGANs [14]. The generator of pix2pix is a U-Net architecture, which is universally used in image-to-image translation. U-Net is a structure that directly connects the encoder layer and the decoder layer through a "skip connection". Through the skip connection, more stable learning compared to a simple encoder–decoder architecture is possible. The discriminator employs a convolutional PatchGAN classifier. The PatchGAN classifies images using patches of a specific size rather than the entire area. This trains the generator to produce more realistic images.

2.4. Deep Neural Network (DNN)

A deep neural network (DNN) is a statistical learning algorithm that imitates human neuron cells. It is an artificial neural network with multiple hidden layers between the input and output. Nodes in each layer receive the nodes from the lower layer as input (x),

multiply the weights (w), add a bias (b), and feed them through an activation function to the nest layer, as shown in Equation (1):

$$y_n = f\left(\sum_i w_i x_i + b\right) \tag{1}$$

There are various types of activation functions but, in this study, the activation functions were ReLU and leaky ReLU. Through training, the back-propagation algorithm optimizes the weights to minimize the loss function. The loss function uses the mean squared error (MSE). The loss function to be minimized is defined as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2}$$

for

$$\text{ReLU} : f(\mathrm{x}) = max(0, x) \tag{3}$$

$$\text{Leaky ReLU} : f(\mathrm{x}) = max(0.01x, x)$$

As a result of training, the output value can converge to the actual value according to the optimization of the weights. The schematic diagram of the DNN is shown in Figure 2.
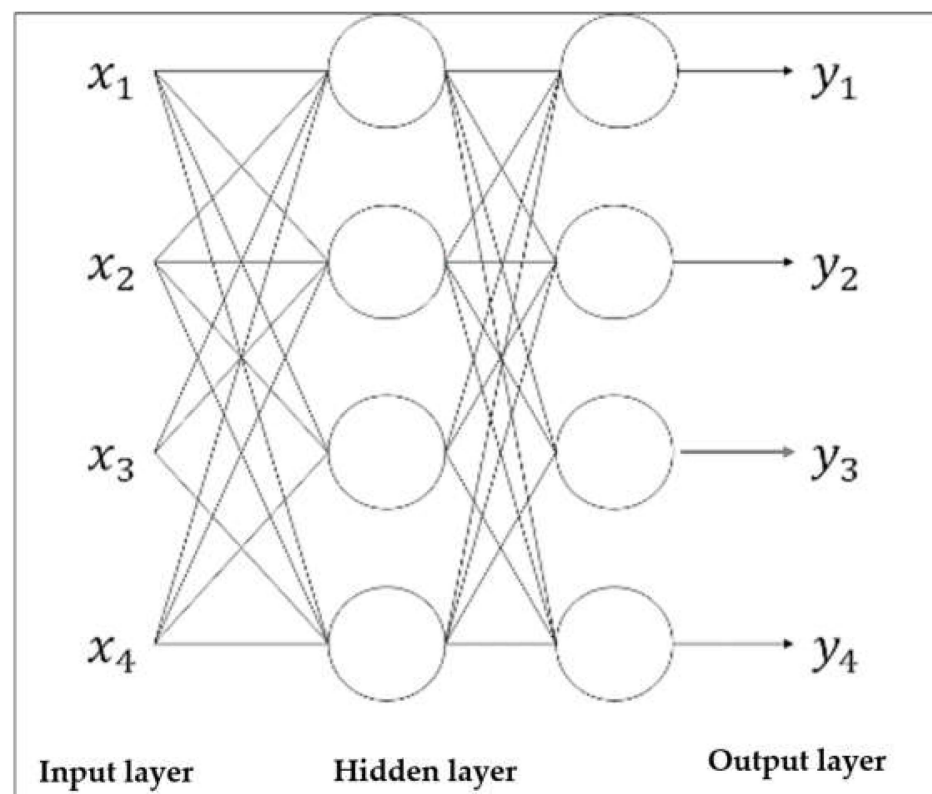

Figure 2. Schematic diagram of the deep neural network (DNN).

2.5. Prediction of Airfoil Flow Field and Aerodynamic Performance Using Pix2pix and the DNN

In this study, pix2pix was used to predict the airfoil flow field. A 19-coordinate image of the airfoil was used as input and the image of the airfoil flow field as the target. Additionally, the angle of attack of the airfoil was displayed as a graph and the Reynolds number was displayed as text. The flow chart for the use of pix2pix in airfoil flow field prediction is shown in Figure 3.
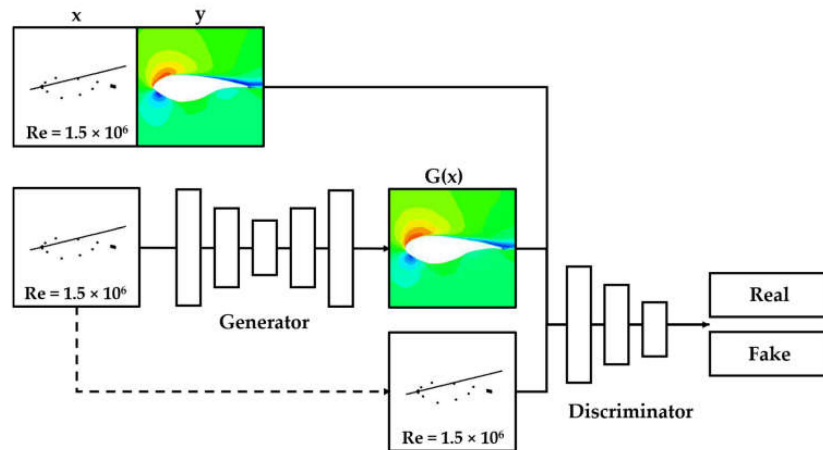
Figure 3. The flow chart for the use of pix2pix in airfoil flow field prediction.

The objective function for training was as shown in Equation (4). L_{cGAN} is a loss function of cGAN, which is optimized toward minimizing the parameter for the generator and maximizing the parameter for the discriminator. In L_{cGAN}, the loss function is the same as Equation (5). L_{L1} is optimized towards minimizing the difference between the actual value (*y*) and predicted value *G(x)*. L_{L1} is the same as Equation (6). λ is the hyper-parameter that balances the L_{cGAN} and L_{L1}.

$$G^* = argminmax L_{cGAN}(G, D) + \lambda L_{L1}(G) \tag{4}$$

$$L_{cGAN}(G, D) = \mathbb{E}_{x,y}[log D(x, y)] + \mathbb{E}_{x,z}[1 - D(x, G(x, z))] \tag{5}$$

$$L_{L1}(G) = \mathbb{E}_{x,y,z}[\| y - G(x, y) \|_1] \tag{6}$$

In this study, a DNN was used to predict the airfoil aerodynamic performance [10]. It used the 19 coordinates of the airfoil, the receiving angle, and the Reynolds number as inputs to predict the coefficient of lift and the coefficient of drag of the airfoil. The structure of the DNN consisted of an input layer receiving 42 inputs and two hidden layers with 84 nodes using leaky ReLU as the activation function and an output layer with 2 nodes to predict the coefficient of lift and coefficient of drag using ReLU as the activation function. The schematic diagram of the implemented DNN is shown in Figure 4.
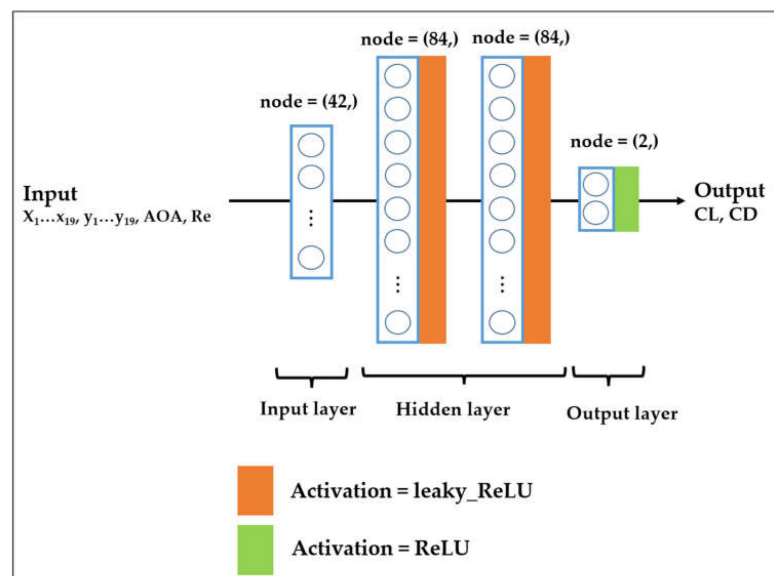

Figure 4. The schematic diagram of the implemented DNN.

2.6. Dataset

To train pix2pix, a dataset was obtained using an in-house CFD code with a genetic algorithm [5]. By applying a genetic algorithm, up to 400 airfoil flow fields of various shapes were obtained for each calculation condition. Simulations were performed with the DU 00-W2-401, DU 00-W2-350, DU 97-W-300, DU 91-W2-250, and DU 93-W-210 airfoils, as shown in Table 1. Simulations were performed with Reynolds numbers of 0.5×10^6, 1.5×10^6, and 3.0×10^6 and with angles of attack of $0°$ to $18°$. The flow fields were obtained using the in-house CFD code developed in [5]. The simulation involved solving the Reynolds-averaged Navier–Stokes (RANS) equations by utilizing the finite volume method, for which the *k-w* turbulence model was employed. The total number of cells was 1.0×10^4, and the computational grid system used is shown in Figure 5. The obtained flow field structure was processed into 256×256 velocity field images using Tecplot.

Table 1. Airfoil CFD calculation conditions. RE, Reynolds number; AOA, angle of attack.

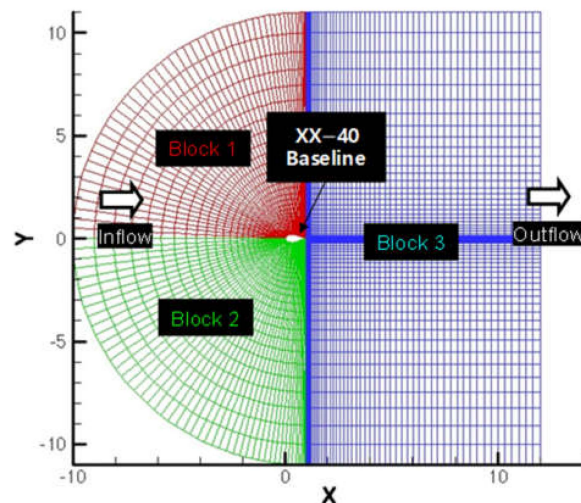|	Condition	Value
Dataset 1	Airfoil	DU 00-W2-401
		DU 00-W2-350
		DU 97-W-300
		DU 91-W2-250
		DU 93-W-210
	RE	1.5×10^6
	AOA	$10°$
	Total number of data points: 606	
Dataset 2	Airfoil	DU 00-W2-401
		DU 00-W2-350
		DU 97-W-300
		DU 91-W2-250
		DU 93-W-210
	RE	0.5×10^6, 1.5×10^6, 3.0×10^6
	AOA	$0°$, $5°$, $10°$, $15°$, $18°$
	Total number of data points: 12,405	

Figure 5. Overall view of computational grid system.

3. Results

3.1. Implementation Details

We constructed two datasets based on the dataset described in Section 2.5. In dataset 1, airfoil flow fields of various shapes were constructed by applying a constant angle of attack and Reynolds numbers of $0°$ and 1.5×10^6, respectively. In dataset 2, five angles of attack and three Reynolds numbers were applied to construct airfoil flow fields of various shapes. Detailed dataset information is shown in Table 1. The dataset was divided into training data, validation data, and test data by dividing the dataset in a ratio of 4:1:1.

3.2. Prediction of the Flow Fields of Airfoils with Different Shapes with Pix2pix

Dataset 1 was trained with a batch size of 2. After the training process of 250 epochs, the mean absolute errors (MAEs) for the training and validation datasets were 0.06602 and 0.1162, respectively. The total learning time was 2 h 22 min. The MAE of the test dataset was 0.1369. The pix2pix model was trained with the adaptive moment (ADAM) optimizer by setting $\beta 1 = 0.5$, $\beta 2 = 0.999$, and $\varepsilon = 10^{-8}$. The initial learning rate was set to 0.0002. Figure 6 shows the pix2pix test results when 19 coordinate images of airfoil shapes were used as input. The left image shows the input and the middle image the flow field obtained through CFD. The right image shows a flow field prediction image obtained through pix2pix. The mean square error (MSE) was applied to quantitatively evaluate the predictive performance. The more similar the image was, the smaller the MSE; for the same image, the MSE was 0. The MSE was obtained using Equation (2). The MSE for the test data was 0.02168. The minimum MSE is usually 0.00551 and the maximum MSE usually 0.09644. The MSEs shown in Figure 6 are: (a) MSE = 0.00702, (b) MSE = 0.00805, and (c) MSE = 0.00874.
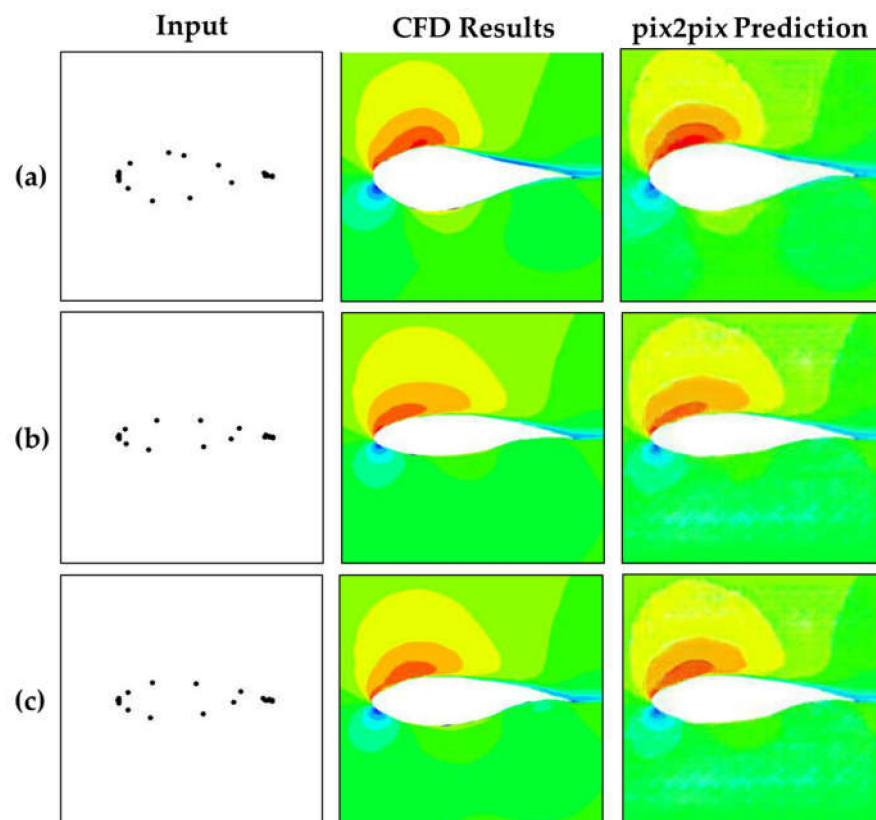

Figure 6. Prediction results of pix2pix for the flow fields of three airfoil shapes with a constant angle of attack ($10°$) and Reynolds number (1.5×10^6): (**a**) DU 97-W-300, (**b**) DU 91-W2-250, and (**c**) DU 91-W2-250.

3.3. Prediction of the Flow Fields of Airfoils with Different Shapes, Angles of Attack, and Reynolds Numbers with Pix2Pix

Dataset 2 was trained with a batch size of 10. After the training process of 50 epochs, the MAEs for the training and validation datasets were 0.06425 and 0.06245, respectively. The total learning time was 7 h and 10 min. The MAE for the test dataset was 0.0764. The pix2pix model was trained with the ADAM optimizer by setting $\beta1 = 0.5$, $\beta2 = 0.999$, and $\varepsilon = 10^{-8}$. The initial learning rate was set to 0.0002. Figure 7 shows the pix2pix test results when 19 coordinates of airfoil images with different angles of attack and Reynold's numbers were used as input. The left image shows the input and the middle image the flow field obtained through CFD. The right image shows a flow field prediction image obtained through pix2pix. The MSE was also determined for dataset 2 for quantitative evaluation. The MSE of the test data was 0.00853. The minimum MSE is usually 0.00251 and the maximum MSE 0.12108. The MSEs shown in Figure 7 are: (a) 0.00531, (b) 0.00358, and (c) 0.00852.
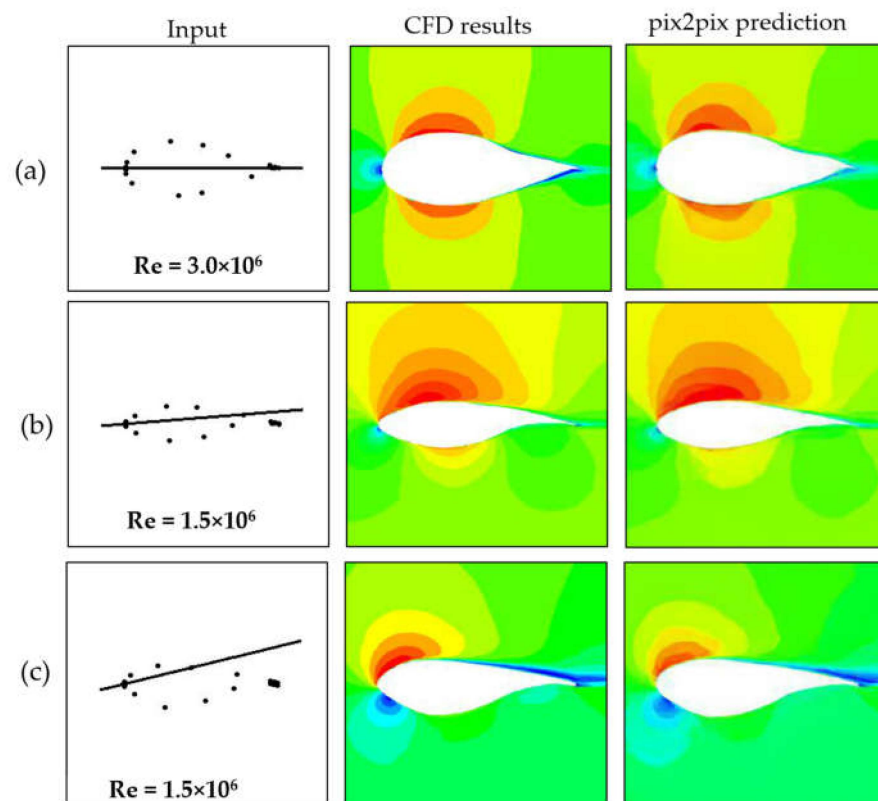

Figure 7. Prediction results of pix2pix for the flow fields of three airfoil shapes at constant angles of attack of 0°, 5°, and 15° and Reynolds numbers of 3.0×10^6, 1.5×10^6, and 1.5×10^6, respectively: (**a**) DU 00-W2-401, (**b**) DU 97-W-250, and (**c**) DU 97-W-250.

Overall, all the results (shown in Figures 4 and 5) were in good agreement with the ground-truth simulation results across the entire ranges of angles of attacks and Reynolds numbers for the three different airfoils.

3.4. Prediction of the Aerodynamic Performance of Airfoils with Different Shapes, Angles of Attack, and Reynolds Numbers with the DNN

The DNN was trained using dataset 2. For training, the batch size was 32 and the number of epochs was 1000. The ADAM optimizer was used to train the DNN model by setting $\beta1 = 0.9$, $\beta2 = 0.999$, and $\varepsilon = 10^{-8}$. The initial learning rate was set to 0.001. The test dataset had a mean square error (MSE) of 0.0087, mean absolute error (MAE) of 0.04, and

mean absolute percentage error (MAPE) of 9.45%. Figure 8 shows that the values predicted by DNN were positively correlated with the true values.
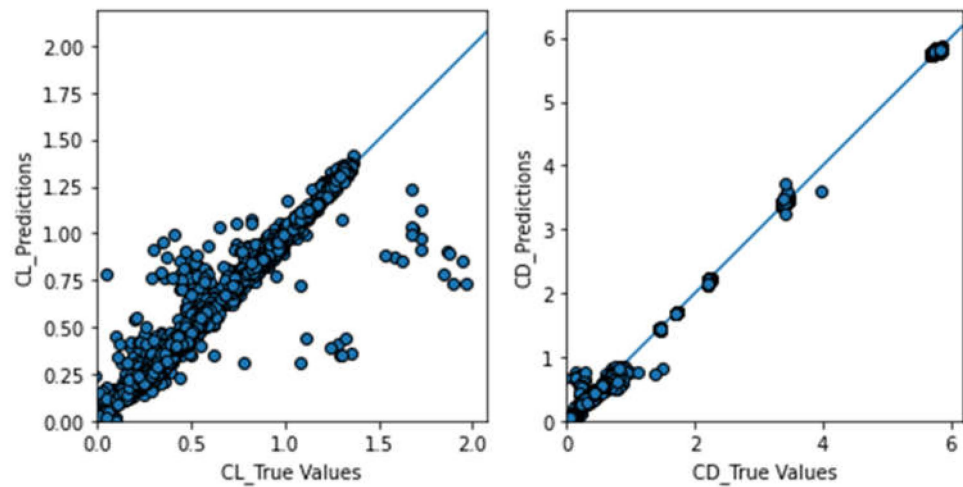

Figure 8. Prediction results with the DNN for the various airfoils.

4. Conclusions

The pix2pix and DNN methods were implemented to predict airfoil flow fields and aerodynamic performance using 19 coordinate images of the airfoil and various Reynolds numbers and angles of attack. The datasets used for the pix2pix and DNN models were established using fully implicit high-resolution scheme-based compressible CFD codes with genetic algorithms. According to the evaluation results, pix2pix was able to predict the flow fields of airfoils, and the DNN was also able to predict the aerodynamic performance of the airfoils. The deep-learning technology established in this paper is proposed as an alternative to CFD for quick identification of the aerodynamic characteristics of airfoils in wind turbine blade design. In future work, we plan to improve the performance of the pix2pix and DNN models and utilize them as wind turbine blade design tools.

Author Contributions: H.-S.S.: investigation, data curation, writing—original draft preparation writing—review and editing. J.M.: validation, data curation, writing—original draft preparation, writing—review and editing, supervision. J.-H.J.: conceptualization, methodology, software, resources, supervision, project administration, funding acquisition. All authors have read and agreed to the published version of the manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare that they have no known competing financial interest or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Global Wind Energy Council. Annual Wind Report, Brussels, Belgium. 2022. Available online: https://gwec.net/global-wind-report-2022/ (accessed on 14 November 2022).
2. Kumar, Y.; Ringenberg, J.; Depuru, S.S.; Devabhaktuni, V.K.; Lee, J.W.; Nikolaidis, E.; Andersen, B.; Afjeh, A. Wind energy: Trends and enabling technologies. *Renew. Sustain. Energy Rev.* **2016**, *53*, 209–224. [CrossRef]
3. Stanley, A.P.J.; Roberts, O.; Lopez, A.; Williams, T.; Barker, A. Turbine scale and siting considerations in wind plant layout optimization and implications for capacity density. *Energy Rep.* **2022**, *8*, 3507–3525. [CrossRef]

4. Buckney, N.; Pirrera, A.; Weaver, P.; Griffith, D.T. Structural Efficiency Analysis of the Sandia 100 m Wind Turbine Blade. In Proceedings of the 32nd ASME Wind Energy Symposium, Harbor, MD, USA, 13–17 January 2014; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2014; pp. 1–27.
5. Jeong, J.-H.; Kim, S.-H. Optimization of thick wind turbine airfoils using a genetic algorithm. *J. Mech. Sci. Technol.* **2018**, *32*, 3191–3199. [CrossRef]
6. Chehouri, A.; Younes, R.; Ilinca, A.; Perron, J. Wind Turbine Design: Multi-Objective Optimization. In *Wind Turbines—Design, Control and Applications*; IntechOpen: London, UK, 2016.
7. Sekar, V.; Jiang, Q.; Shu, C.; Khoo, B.C. Fast flow field prediction over airfoils using deep learning approach. *Phys. Fluids* **2019**, *31*, 057103. [CrossRef]
8. Zuo, K.; Bu, S.; Zhang, W.; Hu, J.; Ye, Z.; Yuan, X. Fast sparse flow field prediction around airfoils via multi-head perceptron based deep learning architecture. *Aerosp. Sci. Technol.* **2022**, *130*, 107942. [CrossRef]
9. Guo, X.; Li, W.; Iorio, F. Convolutional Neural Networks for Steady Flow Approximation. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; ACM: New York, NY, USA, 2016; pp. 481–490.
10. Kutz, J.N. Deep learning in fluid dynamics. *J. Fluid Mech.* **2017**, *814*, 1–4. [CrossRef]
11. Li, J.; Du, X.; Martins, J.R.R.A. Machine learning in aerodynamic shape optimization. *Prog. Aerosp. Sci.* **2022**, *134*, 100849. [CrossRef]
12. Bhatnagar, S.; Afshar, Y.; Pan, S.; Duraisamy, K.; Kaushik, S. Prediction of Aerodynamic Flow Fields Using Convolutional Neural Networks. *Comput. Mech.* **2019**, *64*, 525–545. [CrossRef]
13. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
14. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5967–5976. [CrossRef]
15. Gauthier, J. Conditional Generative Adversarial Nets for Convolutional Face Generation. Computer Science 2015. Available online: https://api.semanticscholar.org/CorpusID:3559987 (accessed on 9 November 2022).
16. Mirza, M.; Osindero, S. Conditional Generative Adversarial Nets. *arXiv* **2014**. [CrossRef]